ISSUE: JUNE 1997

LISTing Policy ..... Annual Dues $16.00

LISTings

Newsletter of the Long Island Sinclair / Timex Users' Group

15 YEARS AND STILL GOING STRONG!

Pres.	Bob Gilder Pro tem
Vice Pres.	Bob Gilder
Tresurer	Robert Malloy
Cor. Secy.	John Pazmino
Assoc. Editor	Fred Stern
Publisher	Bob Gilder
Libr.	Tom Skapinski

Please send all inquiries and submissions (including dues) to: L.I.S.T.
Mr. Bob Gilder
69 Jefferson Place,
Massapequa, N. Y. 11758

COMING EVENTS: The next L.I.S.T. meeting will be Sunday, 9/14/97 at 2 P.M. at the home of Bob Gilder (see address above).

NEXT MEETING: SEPTEMBER 14, 1997

On e sample copy sent upon receipt of business size SASE. Copies provided on Exchange basis with other Bona fide user groups. We are always looking for articles, programs, reviews, etc to keep members informed and entertained. You maintain full credit and copyright.

QL CORNER

I have received a letter from the NEW Quanta Head Librarian:

Roy Breeton
94 Teignmouth Rd.,
Clevedon,
N. Somerset
June 05, 1997

Dear Bob,

Re: Reduction in numbers of Quanta Librarians

It has become clear to the Quanta committee that there has been a big reduction in requests for Library programs over the past 12 - 18 months. After much discussion, it has been decided that, due in no small way to financial constraints, there will be a reduction in the number of Librarians both in the UK and overseas.

It therefore falls upon myself as Head Librarian to perform the onerous task of informing you as of the 1st of July, 1997, your services as Librarian will effectively cease.

I, and the rest of the committee, are very concious of the valuable service that you have performed over the past years, and to this effect, you will receive complimentary copies of the Quanta newsletter until January 1998.

There will be no need for you to return any software/hardware that has been connected with your task as Librarian.

If you have any questions about this decision, please contact myself or any member of the Quanta committee.

Yours

Roy Brereton

Roy Brereton - Secretary / Head Librarian

At first I became peeved at this letter, and within a few hours I realized that, what am I so angry about? Back in 1985 I joined Quanta for the Newsletter and the Quanta library. When Bob Dyl had to let go being the US Librarian and I took over the task of the East coast Librarian. Little did I know what I had got into. First, I had to purchase approximately four hundred 720K, DD diskettes and labels.

During the first year I purchased over 500 diskettes. I sold one whole Library consisting of 109 diskettes. It took quite some time for me to get paid for this task! The disks had to be formatted, then added the labels which were typed. Then they had to be boxed and then mailed at the Post Office. I decided to count all the disks that I had formatted and duplicated program disks for the Quanta Library which totalled 1327 disks for 2 1/2 years.

Now I can relax, play with my QL, work in my garden, enjoy the beach as the summer wears on.

Quanta Magazine - June 1997, Page 10, Library Cornwe, Roy Brerton

Once again, it has fallen upon my shoulders to take up the role of Head Librarian. My predecessor, Vic Avery has successfully dealt with the task of Head Librarian over the past years, and we are indebted to vic for the hours spent at workshops and at home on his duties. I am sure that Vic (and his wife!) can find much better use for their time now and I wish them every success in their retirement years.

A major topic at the first committee meeting of the new Quanta committee was the need for financial constraint.

So far as the Library is concerned took into account that the number of library requests has reduced dramaticically in the past couple of years, and came to the view that the cost of running this facility needed to reflect the current level of usage.

Wilst very appreciative of all time and effort Librarians have put into providing the library service for members over the years nevertheless Committee decided that drastic reduction in the number of Librarians both in the UK and overseas is the most appropriate course to pursue.

The alternative to reducing the number of Librarians would have been to re-introduce charges for library requests, but this was felt to be unacceptable.

When the dust has settled, you will find that the UK is to be served by one Librarian (Norman Dunbar) for normal requests including C programs, and another (John Southern) who will serve the UK and the overseas requests not covered by Eros Forenzi. The European requests will be served by Eros Forenzi in Italy.

The full address and telephone for the Librarians is as follows:

Norman Dumbar
35 ParkFields
23 Park Road
Eccleshill
Bradford
West Yorkshire
BD10 8AN
Tel:01274 632141

John Southern
40 Dustaff Rd
Poynton
Cheshire
SK12 1HN
Tel:01625 850067
e-mail: tarragon
@dircon.co.uk

Eros Forenzi
5 Via Valeriana 44
23010 Berbenno
(Sondrio)
Italy
Tel:+39 342 590450

Software for the Quanta library will go to John Gregory whose address the front cover of the newsletter. His title is now Software Editor which reflects more accurately the demands of his position.

As usual for the Head Librarian, I will be the only source for software such as Kermit, but I will, if necessary, be able to help any member who asks for library programs.

Roy Brerton

Well, now you know what the story is!

See you on the September issue of LIST... *Bob Gilder*

Bob Malloy received the following message through the net from Ron Kneusel. Perhaps one of our members or any readers of LISTing can help Ron get the advise he needs. If you have an extra TS 2068 manual contact Ron at his e.mail address.

To: bmalloy@idt.net
Subject: Timex/Sinclair 2068
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Status: RO
X-Status:

Bob-

I got your email address from John Pazmino. I recently bought a T/S 2068 at Goodwill ($7) and know virtually nothing about it. He told me of your organization, LIST. Do you have a newsletter or web site?

I'm interested in BASIC commands, memory maps, i/o locations, ROM slot info, etc. Just about anything that would let me use the machine. Any carts that might be available as well.

I was told on the net that it needs a 15V, 1 Amp power supply. I woke it up using 12V (6V lantern battery and 4 D-cells) with slightly shaky video. It also came up using a 9.75V, 650 mA power supply for a T/S 1000, but the video was too shaky to be useful.

Any help appreciated! How many of these were made/sold? I'd never heard of it before. The T/S 1000 seem pretty common, but not this one.

- Ron Kneusel
rkneusel@post.its.mcw.edu
From rkneusel@post.its.mcw.edu Tue Jun 24 10:10:21 1997
Received: from post.its.mcw.edu (post.its.mcw.edu [141.106.32.10])
by u2.farm.idt.net (8.8.5/8.8.5) with ESMTP id KAA26110
for <bmalloy@idt.net>; Tue, 24 Jun 1997 10:10:21 -0400 (EDT)
Received: from keaggy.intmed.mcw.edu (jldh449-1.intmed.mcw.edu [141.106.64.19])
by post.its.mcw.edu (8.8.4/8.8.4) with SMTP
id JAA16519 for <bmalloy@chelsea.ios.com>; Tue, 24 Jun 1997 09:10:18 -0500 (CDT)
Posted-Date: Tue, 24 Jun 1997 09:10:18 -0500 (CDT)

This is Ron's street address:

Ron Kneusel
8725 West Burdick Avenue,
Milwaukee, WI 53227

ZX Burglar Alarm

PROJECT

Feeling insecure? Try this neat burglar alarm interface for your ZX81.

By D.C. McMahon

THIS MICROPROCESSOR controlled security system will monitor up to eight remote switch positions each of which can be either normally open or normally closed, allowing you to use any combination of pressure mats, magnetic catches, window foil or other devices as the sensors. When triggered, it immediately sounds an alarm and then displays the number of the affected input on a seven-segment LED. It can be used with the ZX81 and quite aside from its value as an intruder alarm it provides a useful introduction to the techniques of microprocessor interfacing using machine code and the Z80 Parallel Input/Output Controller.

The Z80 PIO consists of two groups of eight lines, port A and port B, and each line can act as either a data input or output. If the IN instruction is used, the data on those lines defined as inputs is loaded into the accumulator. If the OUT instruction is used, data on the accumulator is latched onto the output lines. The PIO can be programmed to act in any one of four modes:

Mode 0 — Output mode
Mode 1 — Input mode
Mode 2 — Bidirectional mode
Mode 3 — Control mode

The first three modes require the use of the handshaking facilities and so for this design the much simpler control mode has been used, allowing easy input and output of data to and from both eight bit ports on the PIO. Port A provides the eight inputs from the sensing switches while port B is split into four inputs and four outputs, the inputs being used to enter data while the outputs feed the seven segment display. An eight bit word is entered via port B into the register which tells the microcomputer which of the input lines should be high and which low (corresponding to normally open and normally closed switches respectively). The contents of this register are then continuously compared with the inputs to port A and if any discrepancy is discovered a '9' is sent out to the display. A timing loop, formed by loading a high number into the registers H, L, and then successively subtracting one until the result is zero, ensures that the nine is displayed for at least a second or so, after which the number of the affected input is displayed. An AND gate monitors the A and D data lines into the display driver and thus goes high when the 9 is output, the delay ensuring that it remains high long enough to latch the relay and thus sound the alarm.

Construction

Most of the components, including the relay and the transformer, are mounted on the PCB, the only off board components beings the switches, the LED display, the edge connector to suit the ZX81, and the siren or other output transducer. Make sure that all four ICs are inserted the right way around, and similarly check the electrolytic capacitors C1, C4, and C7, and the diodes. Provision has been made for the use of connectors for the LED display and the input lines but if you prefer you can, of course, solder

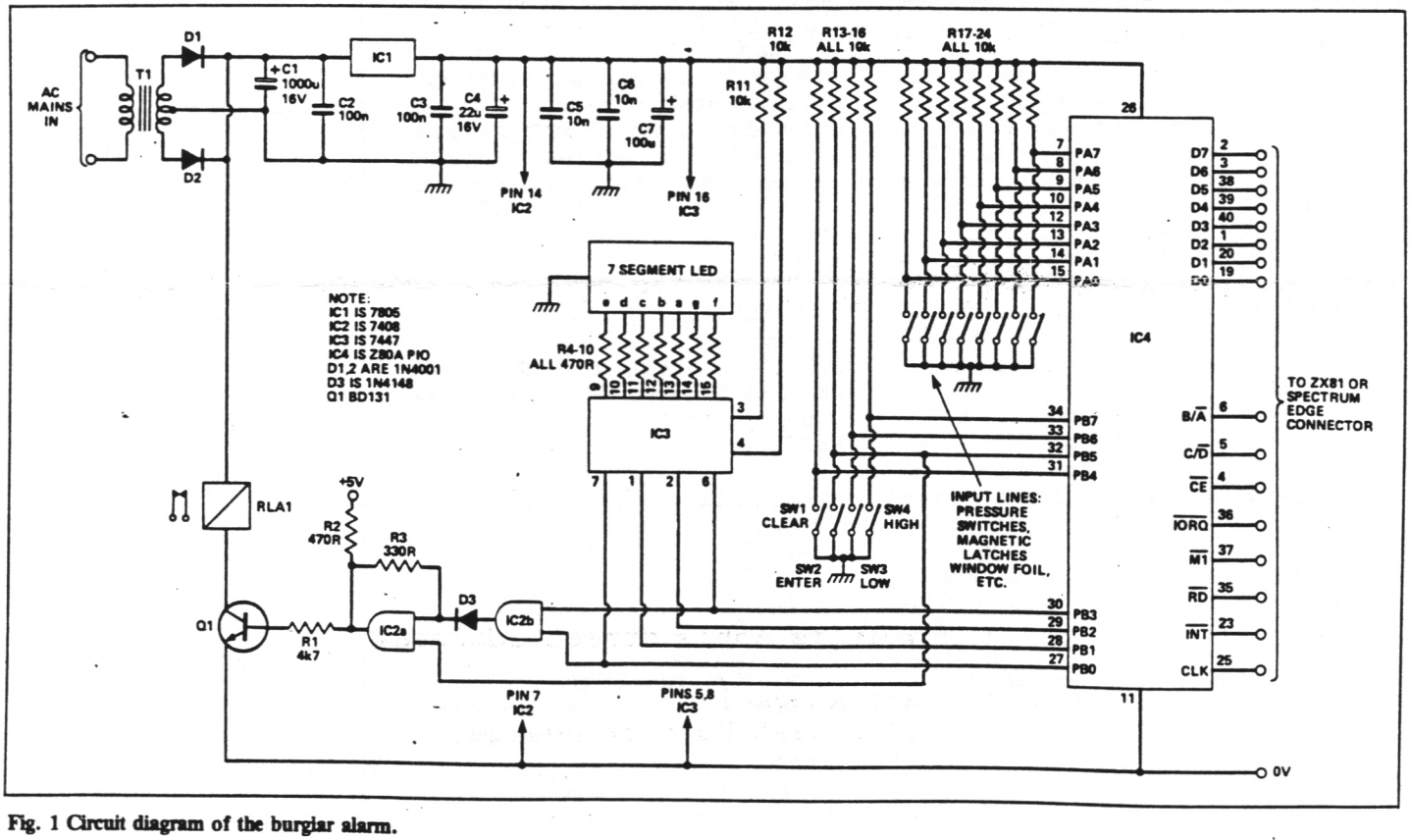


Fig. 1 Circuit diagram of the burglar alarm.

directly to the board. It is intended that the relay should switch a siren or similar device which draws its power from an external source, e.g. the mains, but if your particular application does not specifically demand an eardrum piercing, complaint eliciting siren, you might prefer to use an audible warning device of some sort instead. Providing this does not draw more than 100 mA or so and will run from 17V or less, you can connect it directly in the collector circuit of Q1 and dispense with the relay entirely. The edge connector for the ZX81 should be wired in accordance with Fig. 2.

The choice of case is left entirely to the constructor, but since there is mains on the board it is advisable to have some sort of enclosure. Mounting the switches should present no problems but the LED display is not so easy. If you're after a particularly neat appearance you would perhaps do best to go for an easily cut plastic case, and to cut out an aperture for the LED and then mount it flush in epoxy. The input lines, mains input, and connections to the microcomputer could either be taken through grommets or, if you're really fussy, through appropriate connectors, although it is probably most convenient to use a connector only for the input lines.

Programming

The Z80 PIO has six control lines, three of which (MI, IORQ, and RD) can be connected directly to their counterparts on the ZX81 edge connector. The remaining three, B/A SEL (select port A or B), C/D SEL (select either control or data carried on bus), and CE (chip enable) must be connected to the address bus. The ZX81 address bus has the following characteristics: A0, A1, A2, A3, and A4 are all normally high (they are used to control printer, loudspeaker, etc.), so we can leave these high and connect B/A SEL, C/D SEL, and CE to the remaining three lines, A5, A6, and A7 respectively. A5 low selects port A, A5 high selects port B; A6 low selects data (input and output) and A6 high selects control (programming information). A7, the chip enable, is always held low. The resulting eight bit words are shown in Table 1 and their decimal values are 31, 93, 63, and 127 respectively.

We must next initialize the PIO by sending two control words to each port. The first defines which mode and, as we are using mode three, a second must be sent to define which of the eight lines are inputs and which outputs. The format of the operation control word is shown in Table 2, and it will be seen that the relevant control word for ports A and B is 11111111, that is, decimal 255.

The second control word also consists of eight bits, each one corresponding to the I/O line with the same number, i.e., bit 0 corresponds to PA (or PB) 0, bit 1 corresponds to PA1, etc. Setting the bit high defines the associated I/O line as an input,

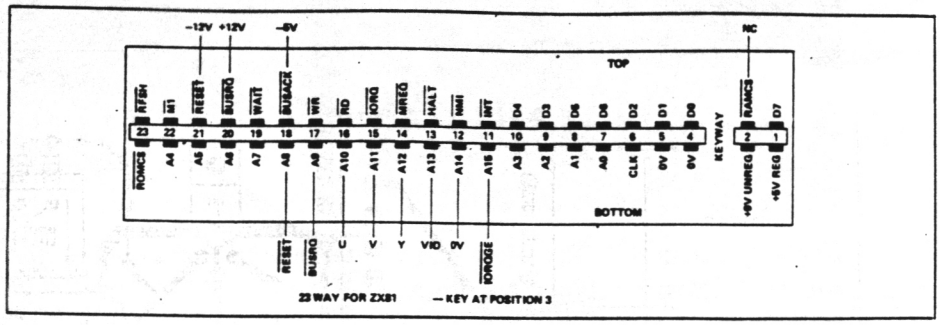


Fig. 2 Pin connections of the ZX81 expansion port

A7	A6	A5	A4	A3	A2	A1	A0	PORT
0	0	0	1	1	1	1	1	A data
0	1	0	1	1	1	1	1	A control
0	0	1	1	1	1	1	1	B data
0	1	1	1	1	1	1	1	B control

Table 1 Examples of eight bit PIO address words.

BIT	7	6	5	4	3	2	1	0
output mode	0	0	1	1	1	1	1	1
input mode	0	1						
bidirectional	1	0						
control	1	1						

Table 2 Format of the operation control word.

while setting it low defines it as an output. Since port A consists of the eight input lines from the various sensing switches, its control word will be 111111111, again, decimal 255. Port B has lines 0, 1, 2, 3 outputting data to the LED display and lines 4, 5, 6, and 7 accepting input data from the push button switches, so its control word will be 11110000, that is, decimal 240.

The first six instructions of the program therefore consist of loading the relevant control word into the accumulator and outputting it to either address 93 or address 127 (see Table 4 and the flow chart, Fig. 3).

Before the progam can be entered, you will need to reserve space for the 108 bytes of machine code by moving RAM-TOP. To do this type in:

```
POKE 16388,147
POKE 16389,67
NEW
```

and follow each statement with the Newline command. To check that RAMTOP has been moved, type in:

```
PRINT    256*PEEK 16389 + PEEK
         16388
```

and you should get 17299.

Having reserved the 108 bytes after RAMTOP in the ZX81, type in:

```
10 FOR N=0 to 107
20 INPUT X
30 POKE 17299+N,X
40 NEXT N
50 PRINT USR 17299
```

then RUN. The computer will then wait for you to type in the 108 numbers given in Table 3.

When the programming has been completed, a 1 should appear on the LED

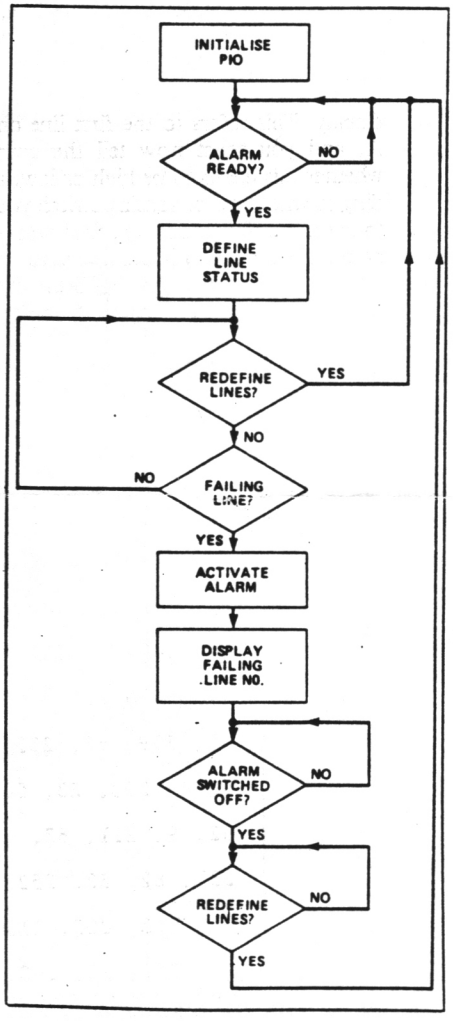


Fig. 3 Flow chart of the burglar alarm program.

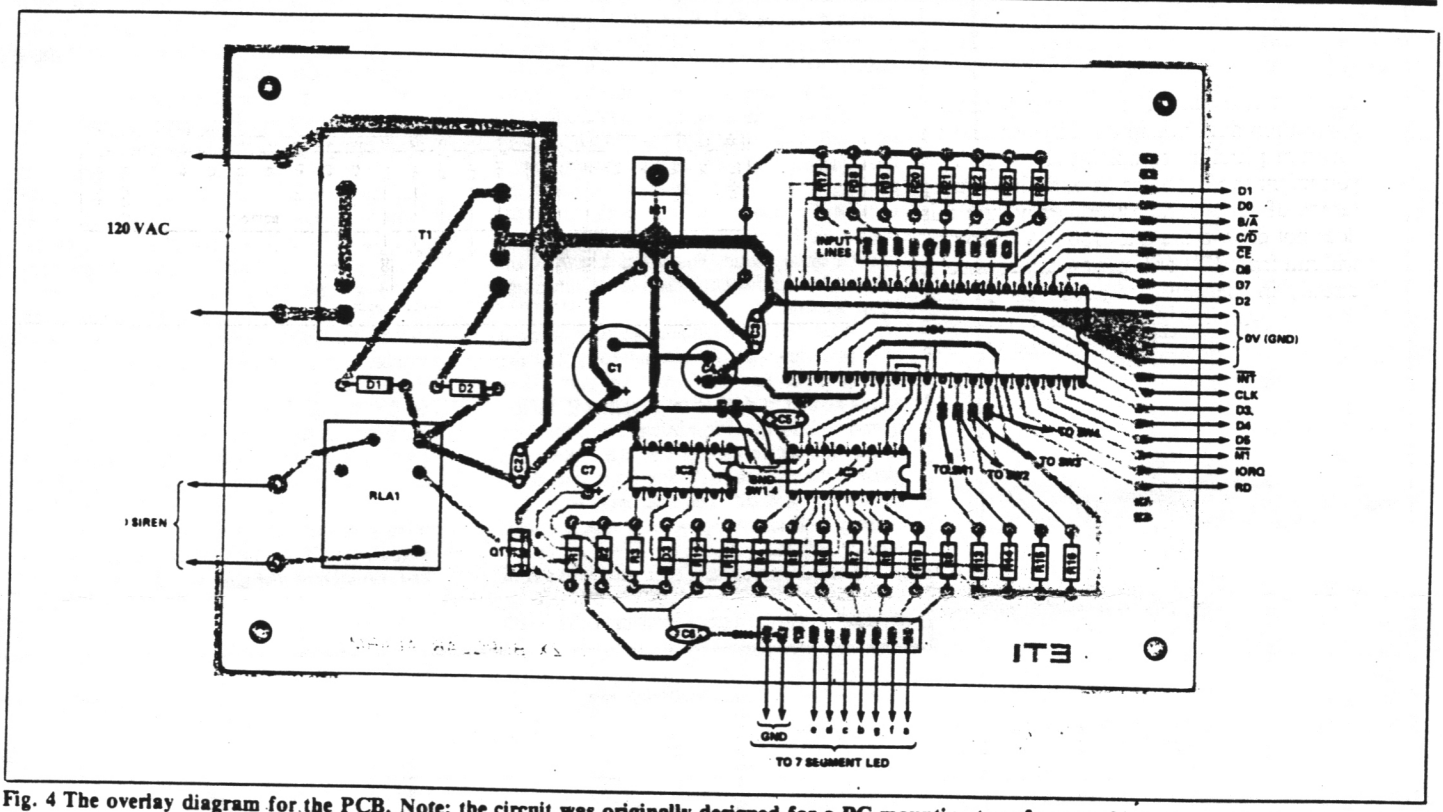


Fig. 4 The overlay diagram for the PCB. Note: the circuit was originally designed for a PC-mounting transformer which may not be available. An ordinary transformer can be mounted off the board with the leads soldered to the PC pads.

display. This refers to the first line on port A, and you must now tell the computer whether this line is to be high or low, according to what type of sensing switch you plan to use on it. To do this you first press either switch SW3 (LOW) if the line is to be normally closed or SW4 (HIGH) if the line is to be normally open, and then press the ENTER switch SW2. The LED should now display a 2, and you repeat the procedure with this and each of the subsequent lines.

When all eight lines have been entered and the register is full, the microprocessor goes into a continuous loop, checking each line against its corresponding bit in the register. Should you wish to redefine the normal state of the lines, pressing SW1 empties the register and thus stops the program. If the alarm is triggered, it can be reset by pressing first SW2 and then SW1.

```
62, 255, 211, 93, 211, 93, 211, 127, 62, 240,
211, 127, 219, 63, 203, 111, 40, 250, 6, 8,
62, 0, 30, 0, 60, 211, 63, 219, 63, 203, 119,
40, 5, 203, 127, 32, 246, 28, 219, 63, 203,
111, 32, 250, 203, 11, 16, 232, 62, 0, 211,
63, 219, 63, 203, 103, 40, 210, 219, 31, 171,
6, 8, 203, 23, 56, 4, 16, 250, 24, 237, 197,
62, 9, 211, 63, 6, 10, 17, 1, 0, 33, 222, 57,
237, 82, 32, 252, 16, 247, 193, 120, 211, 63,
219, 63, 203, 111, 32, 250, 219, 63, 203, 103,
32, 250, 24, 160
```

Table 3 The ZX81 data.

PARTS LIST

Resistors (all ¼ W, 5%)

R1	4k7
R2, 4,5,6, 7,8,9,10	470R
R3	330R
R11, 12, 13, 14 15, 16, 17, 18, 19, 20, 21, 22, 23, 24	10k

Capacitors

C1	1000u 16V electrolytic
C2, 3	100n
C4	22u 16V electrolytic
C5, 6	10n ceramic
C7	100u 10V tantalum

Semiconductors

IC1	7805
IC2	7408
IC3	7447
IC4	Z80A PIO
Q1	BD131
D1, 2	1N4001
D3	1N4148
DISP1	Common anode 7-segment display

Miscellaneous

RLA1	12V DC 400R miniature relay
T1	9-0 9V 6VA transformer
SW1, 2, 3, 4	momentary action, push-to-make

PCB: edge connector to suit ZX81; 10-way 0.1" pitch PCB plug and socket — 2 off each; case, etc. to suit.

```
LD  A      255          62,    255
out A       93         211,     93
out A       93         211,     93
out A      127         211,    127
LD  A      240          62,    240
out A      127         211,    127
IN          63         219,     63
BIT 5                  203,    111
JRZ         -6          40,     -6
LD  B        8           6,      8
LD  A        0          62,      0
LD  E        0          30,      0
INC A                   60,
out A                  211,     63
IN          63         219,     63
BIT 6                  203,    119
JRZ          5          40,      5
BIT 7                  203,    127
JRNZ       -10          32,    -10
INC E                   28,
IN          63         219,     63
BIT 5A                 203,    111
JRNZ        -6          32,     -6
RRE                    203,    111
DJNZ       -24          16,    -24
LD  A        0          62,      0
out         63         211,     63
IN          63         219,     63
BIT                    203,    103
JRZ        -46          40,    -46
IN          31         219,     31
XOR E                  171,
LD  B        8           6,      8
RLA                    203,     23
C  4                    56,      4
DJNZ                    16,     -6
JR         -19          24,    -19
PUSH B                 197
LD  A        9          62,      9
out         63         211,     63
LD  B       10           6,     10
LD  DE       1          17,      1,      0
LD  HL  14814          33,    222,     57
SBC HL, DE             237,     82
JRNZ        -4          32,     -4
DJNZ        -9          16,     -9
POP B                  193
LD A, B                120
out 63                 211,     63
IN  63                 219,     63
BIT 5                  203,    111
JR NZ       -6          32,     -6
IN  63                 219,     63
BIT  4                 203,    103
JRNZ        -6          32,     -6
JR         -96          24,    -96
```

Table 4 Assembler listing of the burglar alarm program.

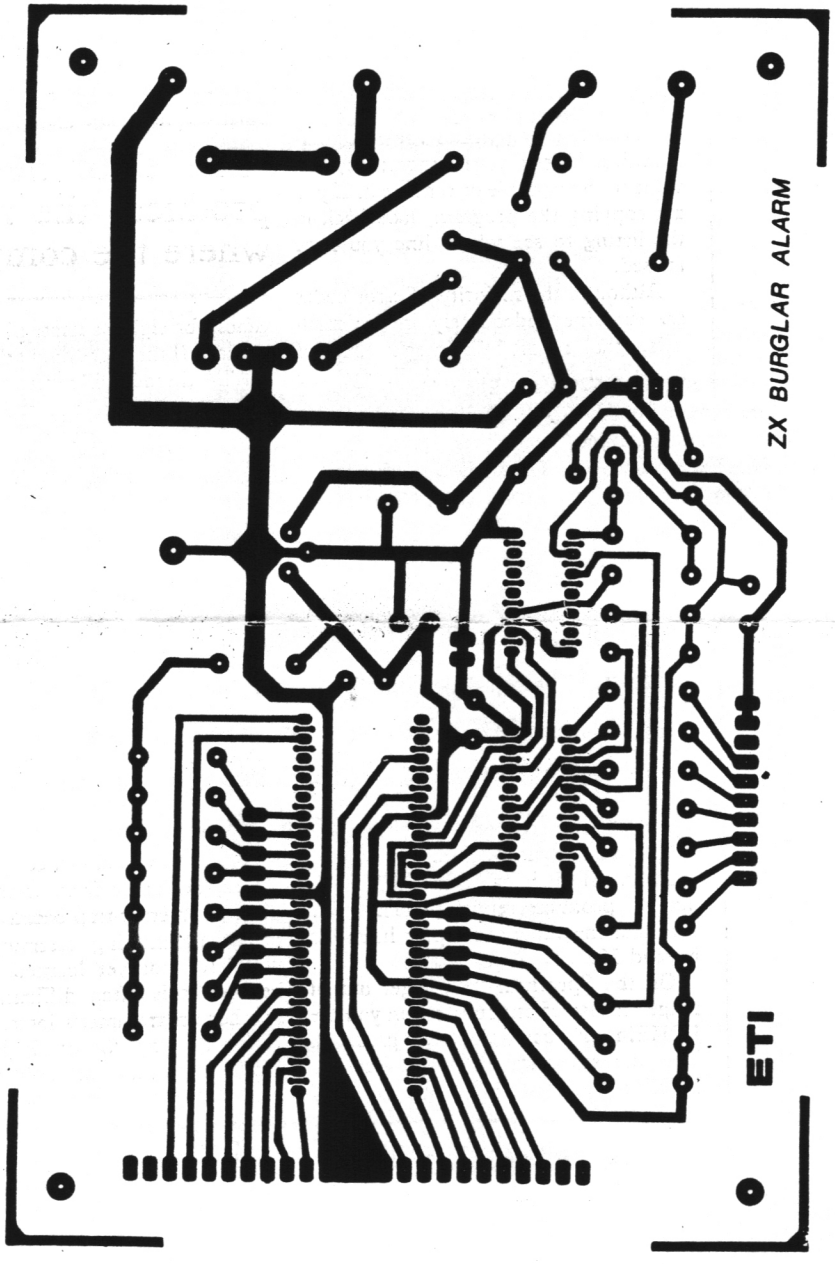
HOW IT WORKS

The various intruder detecting switches are connected between ground and the eight lines of port A on the PIO. Each of the eight lines is connected to the +5V line through a pull-up resistor, so that when the associated switch is open a logic high level will appear on the input, and when the switch is closed the line will be pulled down to logical low. The latter four lines on port B are similarly connected so that pushing any of switches 1 to 4 takes the associated line low. The first four lines on port B are used for the display output and carry a four bit binary code. This is fed directly to the decoder/driver 7447 and then to the seven segment display.

When the program is executed it puts out a '1' and then waits for the line to be defined. Taking either line 6 or line 7 on port B low enters a 0 to 1 as desired into register E. Subsequently taking line 5 low initiates a rotate right instruction which moves the entered data one place to the right so that the register is ready to receive the next bit. The microprocessor then outputs a '2' and the process is repeated until register E is full.

The microprocessor then goes into a continuous loop, using the XOR function to simultaneously compare each input line with the corresponding bit in register E. If both bits are at the same level, either both high or both low, the XOR function will produce a 0 output, but if the two bits are at different logic levels the XOR will give a 1. The RLA instruction is used to shift each bit into the carry flag and test for a 1 and if no carry is detected the microprocessor carries on testing the lines.

When a 1 is detected, a nine is briefly sent out via port B to the display. At the same time, a large number is loaded into registers H and L and 1 is successively subtracted until the result is zero. A total of 148 140 machine cycles are needed for this, and the nine is therefore displayed for a full second or so before the micro processor removes it and displays instead the number of the failed line. The AND gate IC2b has its inputs connected to the A and D lines from port B, and will therefore go high only when a nine is put out. Its output is connected to IC2a, another AND gate, which is wired as a latch. IC2a drives the transistor Q1 which turns on the relay. The other input of IC2a is connected to line 5 of port B, and if SW2 is pressed this line will go low, unlatching the gate and thus turning off the relay.



Trapping the errors will ensure first-time running

THE FIRST error code encountered by most Sinclair users is the flashing "S" on the ZX-81 or "?" on the Spectrum, which indicates a syntax error in a sentence. Experience and the manual soon show that it is caused usually by typing-in a keyword letter by letter, or by bad punctuation, for example omitting a semi-colon or an inverted comma.

The most frequently-occurring error code is "2" — variable not found. A variable is a letter which has been given a numeric value. When you enter "LET a=2" you are defining a variable. Error code 2 results when the computer reaches a variable in the program to which you have so far given no value.

Check the line which the computer specifies. If it is your program, give a value to the variable or remove it. If you are copying the program, look back in the listing to see which line you have missed.

Although the majority of error codes are explained adequately in the manuals, the report "B-Integer out of range" can be confusing. An integer is a whole number — 1 is an integer, 1.5 and 1½ are not. That code occurs most frequently when you try to print something beyond the limits of the screen.

PRINT AT 0,31; "a" is acceptable and will print a letter "a" at the top right of the screen. PRINT AT 0,32; "a" would not be possible. The integer 32 would be out of range, resulting in error code "B". That would also happen if the computer were instructed to PRINT AT 0,31;"ab". It would still be trying to print a character beyond the limits of the screen.

That error is more difficult to detect if variables have been used as co-ordinates and your character, or series of characters, is being printed in varying positions. If the instruction PRINT AT 0,x;"a" produces report code B, make sure that the value of x never increases beyond 31.

On the Spectrum "B-integer out of range" is also often found when you are POKEing-in user defined graphics. The biggest number which can be POKEd-in this case is 255 or BIN 11111111. In that case the error code will occur in the line containing the POKE statement. In most cases, though, the error will have occurred in one of the DATA lines in the program.

A very frequent error code produced on the Spectrum is "E-Out Of Data". That will occur in a line containing a READ statement. The error code, though, will have occurred in one of the program DATA lines, which may be nowhere near the READ line. A READ command sends the computer to a DATA line to collect the next piece of DATA contained there. That is often done using a FOR, NEXT loop, especially when graphics are being set up.

FOR n=1 TO 8: READ n will send the computer to the DATA lines eight times, for eight separate pieces of information. If there are only seven pieces of DATA there it will return to the READ line and produce the code OUT OF DATA. When there are several DATA lines they will all have to be checked, because the piece of DATA you have omitted was not necessarily the last.

> 'The error need not be on the line which produces the report; that is simply the line where the computer meets the problem'

In some cases the computer will follow the program correctly, without producing an error code, but from the programmer's point of view the program contains an error. In that case BREAK into the program at the moment it goes wrong. That will produce report code 9 and the line on which you have STOPped the program. That method makes it easy to locate the area of the program which contains the error.

Programs which you copy from magazines, books or from friends can be difficult to error-trap because they contain programming techniques which you have not yet learned, or simply because it is often difficult to follow another programmer's logic.

The flashing "S" or "?" indicating a syntax error may appear frequently. In that case check carefully what you have copied. You may not have recognised words such as LN or EXP as keywords.

On the ZX-81 especially it is easy to forget that pressing "π" will produce the word PI.

Make sure that when the "is not equal" sign, "<>" appears in a listing you always enter it as one character and not as "is less than", "<" followed by "is greater than", ">".

Technical problems can also cause errors in programs. Any alteration to the power supply can cause a program to CRASH. In that case the screen display may change dramatically and using the keyboard will have no effect. The only solution is to unplug your computer and begin again, making sure that your power supply and RAM pack are both connected firmly. That error is caused by the computer and not by the program.

Sometimes a program listing in a book or magazine will contain what seems to be a very obvious error. If it contains key words or symbols which are not on your computer, check that it is intended for your machine. Programs for the Spectrum, the ZX-81 and the ZX-80 are not usually directly interchangeable. If a program contains the command GOTO or GOSUB — a non-existent line number — the computer will simply go to the next numbered line after that one. That is a sign that a program has been developed and improved and is rarely an error.

When you have errors in a program, first check the report codes listed in Appendix B of the manual. It may then be necessary to read the appropriate section of the manual. Remember that the error is not necessarily on the line which produces the report code; that is simply the line where the computer meets the problem for the first time.

Always check carefully every line connected with the line containing the error code and the mistake should be easy to locate.

ATTENTION LIST Subscribers: When it is time to renew your membership, (look at your mailing label), please make out your check to Harvey Rait, LIST President or to Robert Malloy, Treasurer. PLEASE DO NOT MAKE OUT YOUR CHECK to LIST. Our bank requires a large amount of money in a savings account in order to cash checks. THANK YOU!

Robert Gilder
69 Jefferson Place,
Massapequa, NY 11758

Robert Malloy
412 Pacific Street,
Massapequa Park, NY 11762

Due to rising postage costs outside of the United States, we must raise our annual dues accordingly:

USA postage $16.00
CANADA and MEXICO $17.50 US, and the rest of the world $24.00 US.

Bob Malloy, LIST Treasurer

WHO'S ONLINE

Some of us here at LIST have been wondering how many of our members are using modems with their Sinclair computers. It would be helpful if those of you who are into communications would take a few minutes to let us have the following info.

COMPUTER USED .....
COMMS PRGRM .....
BAUD RATE .....
EMAIL ADDRESS.....
ONLINE SERVICES USED.....
SUGGESTIONS FOR LIST.....

You can reply to me at either of the following addresses:
74776.2342@compuserve.com
bmalloy@chelsea.ios.com (Internet)

Or, you can use our snailmail address.

Bob Malloy

ON LINE

Bob Malloy	74776.2342@compuserve.com
Tom Skapinski	tskapins@juno.com
Jon Pazmino	john.pazmino@moondog.com
Tim Swenson	swensotc@ss2.sews.wpfb.af.mil
Bill Cable	bcable@triton.coat.com
Mike Jonas	mjonas@bbn.com
Gary Norton	gnorton@world.std.com
Al Boehm	boehm@plh.af.mil
Ed Kingsley	elk4@aol.com